Temporal Causal Modeling with Graphical Granger Methods

Andrew Arnold
Machine Learning Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
aarnold@cs.cmu.edu

Yan Liu
IBM T. J. Watson Research
Yorktown Heights, NY 10598
liuya@us.ibm.com

Naoki Abe
IBM T. J. Watson Research
Yorktown Heights, NY 10598
nabe@us.ibm.com

ABSTRACT

The need for mining causality, beyond mere statistical correlations, for real world problems has been recognized widely. Many of these applications naturally involve temporal data, which raises the challenge of how best to leverage the temporal information for causal modeling. Recently graphical modeling with the concept of "Granger causality", based on the intuition that a cause helps predict its effects in the future, has gained attention in many domains involving time series data analysis. With the surge of interest in model selection methodologies for regression, such as the Lasso, as practical alternatives to solving structural learning of graphical models, the question arises whether and how to combine these two notions into a practically viable approach for temporal causal modeling. In this paper, we examine a host of related algorithms that, loosely speaking, fall under the category of graphical Granger methods, and characterize their relative performance from multiple viewpoints. Our experiments show, for instance, that the Lasso algorithm exhibits consistent gain over the canonical pairwise graphical Granger method. We also characterize conditions under which these variants of graphical Granger methods perform well in comparison to other benchmark methods. Finally, we apply these methods to a real world data set involving key performance indicators of corporations, and present some concrete results.

Categories and Subject Descriptors: H.2.8 [Database Management]: Database Applications–*Data Mining*

General Terms: Algorithms, Performance, Design

Keywords: Graphical models, Causal modeling, time series data

1. INTRODUCTION

Statistical modeling and data mining methods are playing an increasingly critical role in real world applications that involve forecasting and prediction. In domains that involve decision making, such as business intelligence applications, however, it is hardly satisfactory to merely discover the statistical correlations that exist in the data. Causal relationships between the *levers*, or variables that are subject to decision making, and the *outcomes*, those that are objects of optimization, need to be established so that the provided insights can be made actionable.

Causal modeling is an area of active research, with rich existing literature. Most notably the framework of Bayesian networks [12, 18, 22, 8], and the related causal networks [14, 25, 1, 20], have been recognized as suitable frameworks to study this issue. There is some very interesting past work that has revealed cases in which causal structure can be determined purely from statistical tests [26], and sometimes computationally efficiently. Still in general, the problem of determining causal structure is considered a major challenge, both computationally and philosophically. There are many cases in which statistical observations alone are not enough to determine the causal structure among a set of variables, and even in cases where it is possible to do so in principle, efficient algorithms are hard to come by.

In many applications of business intelligence and optimization, the data available for analysis often involve time series information. The question of how to leverage the temporal structure present in such data for better understanding of causal structure among the relevant variables thus naturally arises. Indeed, considerable research has been done on causal modeling with time series data [1, 7, 30, 21]. Most past work, however, has focused on the modeling of causal relationship between temporal variables, thus admitting the formulation of the causal modeling problem as that of standard time series statistical modeling.

In the present paper, we address and explore the question of to what extent temporal information present in time series data can assist in the modeling and understanding of the causal structures between time-persistent features, rather than temporal variables. Take, as an illustrative example, the problem of understanding the causal relationship between various key performance indicators (KPI) one may have about a company. For example, one might ask a question such as "Is the stock price of a company causally affected by the inventory turnover ratio of that company?" Emphatically, the question we are asking here is not how much the turnover ratio this quarter will affect the stock price after a fixed period of time, say two quarters from now, but rather whether and to what extent the turnover ratio affects the stock price.

As it turns out, a seminal work in the area of econometrics by the Novel prize winner, Clive Granger, has addressed this

KDD'07 August 12–15, 2007, San Jose, California, USA.

very question [11]. A notion of causality he introduced, appropriately called "Granger causality", presents one possible solution to this question, and is the one that is of particular interest to us in the present context. Granger causality is based on the intuition that a cause helps predict its effects in the future, beyond what is possible with auto-regression. More specifically, a feature x is said to Granger-cause y, if auto-regressive model for y in terms of past values of both x and y is statistically significantly more accurate than that based just on the past values of y.

As it was originally introduced, Granger causality was defined for a pair of features, and the question of how one could apply this notion to the analysis of time series data involving many features was not directly addressed. Pioneered by the work by Eichler et al, there has recently been some interest in combining the notion of Granger causality with graphical models [7]. However, the area is young and many issues of practical significance remain, such as the question of relative accuracy and efficiency of competing methods. Specifically, with increasing interests in applying model selection methodologies for solving structural learning problems for graphical models, it is natural to ask how best to combine the time series specific notion of causality that Granger provides with these new learning techniques to devise a practical approach to temporal causal modeling.

In the present paper, we conduct a systematic empirical investigation as an attempt to start answering such questions. In particular, we consider a number of variants which, loosely speaking, fall under the category of graphical Granger methods, including the canonical exhaustive Granger method and the Lasso-Granger method. We also compare their performance against some benchmark methods for time series analysis, including the vector autoregression (VAR) method [9] and the SIN method [5] tailored to handle time-series analysis.

We attempt to characterize the relative performance of these competing methods, by conducting a host of systematic simulation experiments, in which a number of parameters of interest are varied and their effects on their performance are observed. Specifically, a large number of simulations are randomly generated in which a target time series model is generated, essentially as a VAR model, and the performance of the various methods is examined as a function of various parameters of the simulation. Finally, we apply the proposed method on an actual data set in the domain of corporate KPIs, obtained from the publicly available S & P Compustat database [27], and exhibit some concrete results.

The rest of the paper is organized as follows. In Section 2, we describe the problem formulation as well as the key notion of Granger causality. In Section 3, we describe the various methods considered in our empirical evaluation and discuss their relationship. In Section 4, we describe the techniques used to evaluate the performance of these methods. In particular, the generation process of the target model and the associated parameters which we vary are described. We then present the results of our experimental evaluation in Section 5. We conclude the paper with some remarks and discussions of open issues in Section 6.

2. PRELIMINARIES

In this section we formally present the problem, introducing notation and definitions. We also describe the key notion of "Granger causality".

2.1 The Problem Formulation

In this section, we precisely formulate the causal modeling problem we consider in this paper. We are interested in modeling and characterizing the causal relationship between *features*, x_1,...,x_p. A *feature causal network* is defined as a directed graph over the features, in which each edge is labeled with a natural number called the *lag* of the edge. The semantics of a feature causal network is akin to that of the popular Bayesian networks, but with the underlying premise that an edge necessarily entails causation, analogously to the interpretation of an edge in causal networks [26]. As in Bayesian networks, the lack of an edge between a pair of features does imply that the two features are conditionally independent, given some subset of the other features.

Given a feature causal network, we associate a certain stochastic process that generates time series data with respect to it. In order to define this stochastic process, it is convenient to introduce the notion of *temporal (or lagged) variables* corresponding to each feature x_i. That is, for some predefined *window size* T, we define temporal variables x_i^0,...,x_i^T, corresponding to feature x_i. The stochastic data generation process of a feature causal network is concretely defined by a corresponding graphical model (Bayesian network) over these temporal variables in the following way: If the lag associated with an edge x_i, x_j is k, then we place a directed edge from $x_i^{T-k} \rightarrow x_j^T$. Given the graphical model over the temporal variables, the stochastic process starts by generating an initial sequence of T feature vectors, and subsequently generating, at any given time step, the next feature vector according to the conditional probability density defined by the graphical model over the lagged set of variables, $P(\{x_i^T\}|\{x_i^t\}_{i=1,...,p,t=0,...,T-1})$, where the variables x_i^t for $t = 0, ..., T-1$ are instantiated with the values in the vectors in the last T time steps. Note that this is essentially the way time series data can be obtained using a "unit causal graph" [1], which plays the role of the graphical model mentioned above. Here the conditional density model for an edge set could in principle be an arbitrary statistical model [13, 1], but in all of our experiments, we assume that they are linear Gaussian models [23]. Similarly, we assume that the initial distribution is a linear combination of Gaussians. With these assumptions, the stochastic models associated with causal feature networks are also equivalent to the so-called Vector Auto-Regressive (VAR) models, but it is important to recognize that the notion could make sense for a wider range of model classes.

Given the above definition of a feature causal network, the goal of a causal modeling algorithm is to infer the structure of the feature causal network, given as input time series data generated by its associated stochastic process. Here the structure of the causal network refers to the directed graph over the feature space, usually excluding the lag labels attached to the edges, or the particular statistical models in the associated temporal data generation model. Thus, the performance of a causal modeling algorithm can be measured purely in terms of a measure of similarity between the output or hypothesis graph and the target graph that gave rise to the input data.

Note that what we present here is closely related to the standard formulation of statistical modeling for time series data, but with a slightly different emphasis in its goal. That is, in the usual formulation, the goal is to recover the associated model of temporal data generation, whereas here

the goal is in understanding the underlying causal structure among the features, which is nothing but the graph structure among the features.

2.2 Granger Causality

As it turns out, a notion of causality that is highly relevant to the present context of temporal causal modeling called "Granger Causality" has been introduced in the area of econometrics [11]. This notion is based on the idea that a cause should be helpful in predicting the future effects, beyond what can be predicted solely based on their own past values. More specifically, a time series (or a "feature" in the terminology of the present paper) x is said to "Granger cause" another time series y, if and only if regressing for y in terms of both past values of y and x is statistically significantly more accurate than doing so with past values of y only. Let $\{x_t\}_{t=1}^{T}$ be lagged variables of x and $\{y_t\}_{t=1}^{T}$ for y, and let $\vec{x_t}$ denote, in general, the vector $\langle x_t \rangle_{t=1}^{t}$. Then, the Granger test is performed by first conducting the following regressions:

$$y_t \approx A \cdot \vec{y_{t-1}} + B \cdot \vec{x_{t-1}} \quad (1)$$

$$y_t \approx A \cdot \vec{y_{t-1}} \quad (2)$$

and then applying an F-test (or some other similar test) to obtain a p-value for whether or not (1) results in a better regression model than with (2) with statistically significantly advantage.

It should be noted that the original notion of Granger causality was formulated in terms of linear regression, but this need not necessarily be the case – there are some non-linear extensions in the literature [2]. It is also important to note that Granger causality attempts to capture an interesting aspect of causality, but certainly is not meant to capture all. In particular, it has little to say about situations in which there is a hidden common cause for the two features in question. More generally, in the present paper, we do not address the important but challenging issue of dealing with hidden variables.

3. METHODS CONSIDERED

In this section we describe the various methods for temporal causal modeling we consider in our experiments, and discuss some of their properties.

3.1 Exhaustive Graphical Granger Method

A natural way of applying the notion of Granger Causality to modeling of time series data involving many features is to simply apply the Granger causality test to each pair of features to determine the presence/absence and orientation of the corresponding edge in the output feature causal graph. Schematically, this method can be represented as follows.

Procedure $Exhaustive\text{-}Granger(X, T)$

1. $X^{lag} \leftarrow \text{Lag}(X, T)$
2. $G = \langle V, E \rangle \leftarrow FullyConnectedFeatureGraph(X)$
3. Determine $G_{feature}^{causal}$ as follows:
   (a) For each edge (x, y) in E,
      i. orient (x, y) as $x \rightarrow y$ if Granger(x, y, X^{lag}) = 'yes' and Granger(y, x) = 'no'.
      ii. orient (x, y) as $x \leftarrow y$ if Granger(x, y, X^{lag}) = 'no' and Granger(y, x) = 'yes'.
      iii. Place an unoriented edge (x, y), i.e. $y \leftrightarrow x$, if Granger(x, y, X^{lag}) = 'yes' and Granger(y, x) = 'yes'.
      iv. Place no edge between (x, y), otherwise.
4. Return$(G_{feature}^{causal})$

Note here that we use Lag(X, T) to denote the lagged version of data X, that is, the data set constructed by appropriately displacing and repeating the *temporal variables* x_i^{lag} = $x_{i,0}...x_{i,T}$ of our original features x_i. $FullyConnected$ $FeatureGraph(X)$ denotes the fully connected graph defined over the features. Also note that we use Granger$(x, y,$ $X^{lag})$ to denote the outcome ('yes' or 'no') of the Granger causality test between features x and y applied on the lagged data X^{lag}, possibly parameterized by a significance level, as described above.

In our experiments, we make use of the "grangertest" function in the lmtest library in R for performing the Granger Test.

3.2 The Lasso Granger Method

The Exhaustive Granger method of the last subsection does not address the issue of combinatorial explosion, both in the computational and statistical senses. Computationally, having to conduct Granger Test, which itself involves applying regression on the lagged variables, $O(p^2)$ times, where p is the number of features, can be prohibitive for large values of p. Also, the statistical significance tests, for all pairs of features, are conducted sequentially without regard to the possible interactions between them.

The Lasso Granger method we consider next is one way to address such issues. One can apply regression to the neighborhood selection problem for any particular feature, namely that of identifying the subset of features on which the feature in question is conditionally dependent, given the fact that the best regressor for that variable with the least squared error will, in theory, have non-zero coefficients only for the variables in the neighborhood.

The Lasso algorithm for linear regression is an incremental algorithm that embodies a method of variable selection using the L1-penalty term [29]. That is, its output, $\vec{w}$, minimizes the sum of the average squared error of regressing for y, plus a constant times the L1-norm of the coefficients, namely,

$$\vec{w} = \arg\min \frac{1}{n} \sum_{(\vec{x},y) \in S} |\vec{w} \cdot \vec{x} - y|^2 + \lambda ||\vec{w}||_1 \quad (3)$$

where S is the input sample, n is the number of examples in S, and λ is a constant to be determined. It is well-known that the L1-penalized least square regression, as targeted by the Lasso, is a convex problem, making it possible to attain the global maximum, via the so-called "least angle regression" procedure, which incrementally updates the weight for one variable at a time [6].

The following summarizes the Lasso-Granger method just described. Note here that we denote by $Lasso(y, X^{lag})$ the set of temporal variables receiving a non-zero coefficient by the Lasso algorithm, when regressing y_t in terms of the lagged variables $x_{t'}, t' = t - T, ..., t - 1$ for all $x \in X$.

Procedure *Lasso-Granger*(X, T)

1. $X^{lag} \leftarrow \text{Lag}(X, T)$
2. $G = \langle V, E \rangle \leftarrow FullyConnectedFeatureGraph(X)$
3. Determine $G^{causal}_{feature}$ as follows.
   (a) For each feature y in V, $w^y = Lasso(y, X^{lag})$.
      i. For each edge (x, y) in E,
         A. orient (x, y) as $x \rightarrow y$ if $x_t \in w^y$ for some t but $y_t \notin w^x$ for all t.
         B. orient (x, y) as $y \rightarrow x$ if $y_t \in w^x$ for some t but $x_t \notin w^y$ for all t.
         C. Place an unoriented edge (x, y), i.e. $y \leftrightarrow x$, if $x_t \in w^y$ for some t and $y_t \in w^x$ for some t.
         D. Place no edge between (x, y), otherwise.
4. Return$(G^{causal}_{feature})$

One question that arises is how to set the parameter λ. In this paper we tried two methods. The first (*lasso time series*) uses the generalized cross validation score [10] (a popular tool for calculating the parameters of regularized linear regression) to select a set of candidate features, and does another round of linear regression to select the most significant subset of these candidates. The second method (*lasso lambda* or *modified lasso time series*) sets λ as in [19]. For completeness, we also tested a non-Grangerized version of lasso (without lagging) called *lasso standard*.

3.3 The SIN Granger Method

As one of the "baseline" methods, we consider the "SIN" method. SIN is a method for structure learning which works very well for linear Gaussian graphical models with relatively small numbers of features, thus it should serve as a good upper bound of ideal performance for that portion of the problem space.

The SIN method rests on the observation that there is no causal relationship between two variables, x_i and x_j, if there exists a subset of variables $X_s \in X \setminus \{x_i, x_j\}$ conditioned upon which x_i and x_j are independent (the so-called assumption of faithfulness [22, 26]). Indeed, this is the main idea behind many causal discovery algorithms such as the PC-algorithm [26, 1, 25].

More specifically, SIN is based on the fact that the d-separation, or conditional independence in graphical models, coincides with the notion of partial correlations. The partial correlation, $\rho_{xy.V}$, between two features x, y is the correlation between them in the conditional distribution given the rest V of the variables. A key fact is that the partial correlation can be computed in terms of the inverse of the covariance matrix Σ^{-1}, known as the concentration matrix, i.e.

$$\rho_{xy.V} = \frac{-\sigma^{x,y}}{\sqrt{\sigma^{x,x}\sigma^{y,y}}} \tag{4}$$

where $\sigma^{x,y}$ denotes the x, y-th element of Σ^{-1}.

SIN is also distinguished by the way it applies "simultaneous" statistical tests on the hypotheses "$\rho_{xy.V} = 0$?", for all pairs x, y, in such a way that the overall error rate can be controlled.

Given the neighborhood determination made by the SIN method, what remains is the orientation of the edges in its output graph. Once again, we resort to the disjunctive collapsing procedure of a variable space graph to the corresponding feature graph, by judging that a directed edge $x \rightarrow y$ is to be placed if there is an edge from at least one of x's lagged variables $x_1, ..., x_{t-1}$ to y_t and vice-versa, and an undirected one if both are true. A schematic representation of the resulting method, SIN-Granger, is given below.

Procedure *SIN-Granger*(X, T)

1. $X^{lag} \leftarrow \text{Lag}(X, T)$
2. $G \leftarrow FullyConnectedVariableGraph(X^{lag})$
3. $G^{undirected}_{variable} \leftarrow SIN(G)$.
4. $G^{causal}_{feature} \leftarrow ProjectFeatures(G^{undirected}_{variable})$
5. Return$(G^{causal}_{feature})$

Note that $FullyConnectedVariableGraph(X^{lag})$ denotes the fully connected graph defined over the lagged variables, as the name implies. The graph in the temporal variable space in Line 3 is projected to the feature space by the $ProjectFeatures$ procedure in Line 4, by merging all the variables x^{lag}_i of a feature x_i into a single node, using the disjunctive semantics described above, and implicitly employed by the previous two methods.

When the covariance matrix can be inverted feasibly, the SIN method does provide a nearly perfect solution to the structure learning for linear Gaussian graphical models, because of the correspondence between the zeros in the concentration matrix and the d-separation in a linear Gaussian graphical model. There are issues, however, for example when the number of features is large and the inversion of the covariance matrix can fail due to under-specification. Also, the computation time can be an issue, due to the nearly cubic time complexity of the matrix inversion process.

In our experiments, we make use of the SIN library in R, to perform the SIN part.

3.4 Vector Auto-Regressive (VAR) Method

As another "baseline" method, we also consider the Vector Auto-Regressive (VAR) model estimation method, which generalizes the univariate auto-regressive (AR) model to multiple time-series. In the simple AR model, an observation of time t is given by

$$x_t = c + \sum_{i=t-T}^{t-1} a_i x_i + \varepsilon_t$$

where a_i is the parameters of the model and ε_t is the Gaussian noise. The stochastic model associated with our causal feature graph is nothing but a VAR model on the lagged temporal variables, and given the assumption that each of the models is a linear Gaussian model, they can be formulated as follows. Letting $\vec{X}_t$ denote the vector of all features at time t, a VAR model is defined as:

$$\vec{X}_t \approx A_{t-1} \cdot \vec{X}_{t-1} + ... + A_{t-T} \cdot \vec{X}_{t-T} \tag{5}$$

where each of the A matrices are $p \times p$ coefficient matrices. This formulation is essentially a notational shorthand for multiple linear regression formulations, one for each of the features x.

The VAR model estimation method is to invert the A matrices in the above formulation, and is basically solving least squared regression problems. In our experiments, we use the "estVARXar" function in the DSE library of R, which in turn makes calls to the "ar" function, an auto-regressive modeling procedure with ARMA, and has a number of distinguishing features: It can handle endogenous variables, optionally invoking a model selection method based on AIC, and it is using the Yule-Walker approach, and thus the modeling is done effectively with the means subtracted out.

Since the data generation process we use is indeed a VAR model, this procedure is also expected to work well and provide another baseline.

3.5 Regularized VAR methods for sparse data

As we pointed out earlier, the estimation using VAR performs well only in cases where the sample size n is much larger than the number of features p, i.e. $n >> p$. Remedy for sparse data can be found in various methods which could be viewed as regularized versions of VAR. Here are a few examples.

- Stepwise variable selection, as adapted in TETRAD [28]. This method is not consistent, however, namely even when the sample size n goes to infinity, it is not guaranteed that the correct set of non-zero coefficients will be selected.

- Stochastic search variable selection (SSVS). SSVS can be thought of as a Bayesian version of the lasso, in which the parameter estimation is explored using Monte Carlo-Markov chain sampling. The use of MCMC limits the applicability of this method to relatively small numbers of features p, as analyzed in [4].

- VAR with Ridge regression is able to achieve stable and plausible estimates when the number of features is much larger than the sample size, i.e. $p >> n$ and demonstrate encouraging improvement in the applications to the brain function prediction [30].

- James-Stein type shrinkage replaces the VAR regression covariance with the Jaimes-Stein shrinkage covariance and is applied to solve problems in system biology [21].

Several comparison studies on the relative performance of different regularization methods show that: in general a hard threshold performs slightly worse than other approaches; in terms of connectivity, ridge regression works the best for graphs with large connectivity while Lasso outperforms others for graphs with small connectivity; in terms of small sample size, Lasso tends to perform poorly and James-Stein type shrinkage works well.[1] Overall, however, the difference in performance between the various penalization schemes is relatively small [21, 30]. Therefore in the present paper we elect to employ the VAR algorithm combined with AIC as a representative of this class of methods.

[1]Note that Lasso applied as regularization on VAR here is to be disntinguished from Lasso Granger of the last subsection – in Lasso Granger, Lasso is applied for regressing each variable, whereas here the L1-penalization is to be applied for the VAR estimation process for the entire vector.

3.6 Consistency of Lasso Granger

One of the major advantages of using Lasso for graphical model structure learning is its consistency. It has been shown that the probability of Lasso falsely including any of the non-neighboring variables of a given node into its neighborhood estimate vanishes exponentially fast, even if the number of non-neighboring variables may grow very rapidly with the number of observations. More rigorously, letting p denote the number of features, a be an arbitrary node in the true graph G, and ne_a be the set of neighbors of a in $G = \langle V, E\rangle$, and θ^{a,ne_a} be the coefficient vector of the optimal linear regressor for a using ne_a, we have the following theorem due to [19].

THEOREM 1. *Suppose the following assumptions are fulfilled:*
(1)high dimensionality: there exists $\gamma > 0$, so that $p = O(n^r)$ for $n \to \infty$. (2) non-singularity: for all $a \in V$ and $n \in N$, $Var(a) = 1$ and there exists $v^2 > 0$, so that $Var(a|V \setminus \{a\}) \geq v^2$. (3) sparsity: there exists $0 \leq \kappa \leq 1$, $\max_a |ne_a| = O(n^\kappa)$, for $n \to \infty$; (4) sparsity: there exists $\vartheta < \infty$, $||\theta^{a,ne_b\setminus\{a\}}||_1 \leq \vartheta$ for all $(a,b) \in E$; (5) magnitude of partial correlations: There exists a constant $\delta > 0$ and some $\xi > \kappa$, such that $\pi_{ab} \leq \delta n^{-(1-\xi)/2}$ for all $(a,b) \in E$, where π_{ab} is the partial correlation between a and b [17]: (6) neighborhood stability: there exists some $\delta < 1$, $|S_a(b)| < \delta$ for all $(a,b) \in E$, where $S_a(b) = \sum_{k \in ne_a} sign(\theta_k^{a,ne_a})\theta_k^{b,ne_a}$. Then, if the penalty for sample size n satisfies $\lambda_n \sim dn^{-(1-\epsilon)/2}$ with some $\kappa \leq \epsilon \leq \xi$ and $d > 0$, there exists some $c > 0$ such that for all $a \in V$ it holds that the estimated neighborhood $\widehat{ne}_a$ satisfies $P(\widehat{ne}_a \subseteq ne_a) = 1 - O(\exp(-cn^\epsilon))$ for $n \to \infty$. In addition, we also have $P(ne_a \subseteq \widehat{ne}_a) = 1 - O(\exp(-cn^\epsilon))$ for $n \to \infty$.

Notice that Theorem 1 holds even for cases in which the number of variables is larger than the number of observations, i.e. $p >> n$.

In our present context, this theorem can be directly applied to derive a corollary on the consistency of Granger Lasso.

COROLLARY 1. *Suppose that a true feature causal graph G and its associated stochastic model (graphical model) M gives rise to time series data. If the assumptions in Theorem 1 are fulfilled by the M's, then Granger Lasso, taking the time series data as input, will output graph which is consistent with the true graph G with probability converging to 1, as n and p tend to ∞.*

PROOF. In step 3(a) of Procedure *Lasso-Granger*, following Theorem 1 we have $P(\widehat{ne}_x^\lambda \subseteq ne_x) = 1 - O(\exp(-cn^\epsilon))$ for $n \to \infty$ and $P(ne_x \subseteq \widehat{ne}_x^\lambda) = 1 - O(\exp(-cn^\epsilon))$ for $n \to \infty$. Given the correctness of the estimated graphical model on the lagged variables, the disjunctive projection in steps A-D in Procedure *Lasso-Granger* will also be correct with respect to the edge presence and orientation in the original feature causal graph. □

3.7 Complexity Considerations

While most research on structure learning focuses on recovering the true graph that gave rise to the data, in many real world applications we need to deal with large-scale data with hundreds or thousands of features. This may prohibit the use of accurate but computationally demanding

methods. The computational complexity is therefore an important factor that directly influences the applicability of a learning algorithm. Here we analyze the computational complexity of the main methods considered in this paper:

SIN. The most computationally expensive operation in the SIN algorithm is matrix multiplication and inversion. As is well-known, the general complexity of inverting a matrix is essentially cubic in the dimension[2]. Therefore the computational complexity of SIN is $O((pT)^3 + n(pT)^2)$

Lasso Granger. Solving the original objective function with L_1 regularizer in Lasso requires quadratic programming, an NP-hard problem in general. However, by rewriting the objective function, we can solve the problem using the Least Angle Regression (LARS), which computes all possible Lasso estimates for a given problem and enjoys a much smaller computational complexity [6]. With this implementation, the computational complexity of Lasso Granger is significantly reduced to $O(n(pT)^2)$.

Exhaustive Granger. The exhaustive Granger tests the Granger causality via regression between each pair of features. Therefore the complexity is $O(n^2p^2T^2)$.

VAR. The VAR formulation is essentially multiple linear regression, which theoretically involves cubic complexity. However, there are many types of speeding algorithms, such as Yule-Walker approach, and the time complexity is between $O(p^2T^2)$ and $O(pT)$.

4. EVALUATION METHODS

In this section we introduce the techniques used to evaluate the temporal causal modeling algorithms laid out in the previous section. Specifically, we describe the data generation process used in our simulation experiments, as well as some measures of graph similarity used to quantify the quality of the models output by the respective algorithms.

4.1 Synthetic Data Generation

The data generation process we use essentially parallels the problem formulation presented in Section 2.1, in terms of the feature causal network and its associated stochastic data generation process. We begin by randomly generating a feature causal network. This random generation is governed by a parameter which we call *affinity*, or the probability of forming a link between any given pair of feature nodes. Having formed the feature causal graph, we then generate a unit causal graph in the temporal variable space that is consistent with it. This is done by randomly choosing the *lag* k for any edge $x \to y$ in the feature causal graph, according to a uniform distribution within a prescribed range and then forming an edge $x^{T-k} \to y^T$ in the unit causal graph.

Once this graph structure is created, we then randomly assign those links some weight, sampled from a specified range(min, max effect), which determines the parameter of the corresponding linear gaussian model. Each variable also gets some gaussian noise of mean 0 and some specified range of standard deviations. We then apply the unit causal graph recursively to obtain the time series data, each of some fixed time steps (Max_T). Once we reach Max_T, we have one complete sample. We then repeat this process to get n samples.

[2]The complexity of matrix inversion can be reduced, for example to $O(p^{2.376})$ by using the Coppersmith Winograd algorithm [3].

4.2 Evaluating Graph Similarity

We now describe how we quantify the similarity between the target causal graph used to generate the input data and the output causal graph. For this, we simply apply the metrics of Precision, Recall and F_1-measure, commonly used in the machine learning and information retrieval literature, to the problem of predicting the 0,1-label in the adjacency matrix representation of the graph. (See, for example, [25] for use of these metrics in evaluation of structural learning methods.) Note that for any pair of features, x_i and x_j, there are two entries in the adjacency matrix A, $A[i,j]$ and $A[j,i]$, each representing an edge going in one direction. A bi-directed edge corresponds to having 1 in both entries, whereas a directed edge would have 1 in one of the entires and a 0 in the other. Given this formulation, precision and recall are well-defined. For example, predicting a bi-directional edge between x_i and x_j, when there is actually a directed edge from $x_i \to x_j$, would entail one correct prediction and one prediction error.

So, letting A^* denote the target adjacency matrix, $\widehat{A}$ the output adjacency matrix, and $V \times V$ the set of feature pairs, *precision* P and *recall* R are defined as follows:

$$P = \frac{|\{(i,j) \in V \times V : \widehat{A}[i,j] = A^*[i,j]\}|}{|\{(i,j) \in V \times V : \widehat{A}[i,j] = 1\}|}$$

$$R = \frac{|\{(i,j) \in V \times V : \widehat{A}[i,j] = A^*[i,j]\}|}{|\{(i,j) \in V \times V : A^*[i,j] = 1\}|}$$

Furthermore, given precision P and recall R, the F_1-measure, F_1, is defined in the usual manner.

$$F_1 = \frac{2PR}{(P+R)}$$

There is clearly a trade-off between precision and recall as the goal of prediction, and the F_1-measure tries to balance the overall quality of prediction.

5. EXPERIMENTAL RESULTS

In this section, we present the results of our experimental evaluation. First, we examine how the relative performance of the competing methods depends on various parameters of the problem. We then present some concrete examples of applying these same methods on a real world data set and discuss their relative merits.

5.1 Synthetic Data

A series of simulation experiments were conducted, each of which consisted of a large number of randomized simulation runs, in which various aspects of the simulation or the problem space were varied, which are mostly parameters of the target causal and stochastic model to be learned from generated data. In each such run, all or a subset of the competing methods were run to obtain their learned models, which were then compared and evaluated against the target model that generated that run's data.

Several series of such experimentation were run: First, all the methods were run on a representative range of the problem space varying all the parameters of the problem space

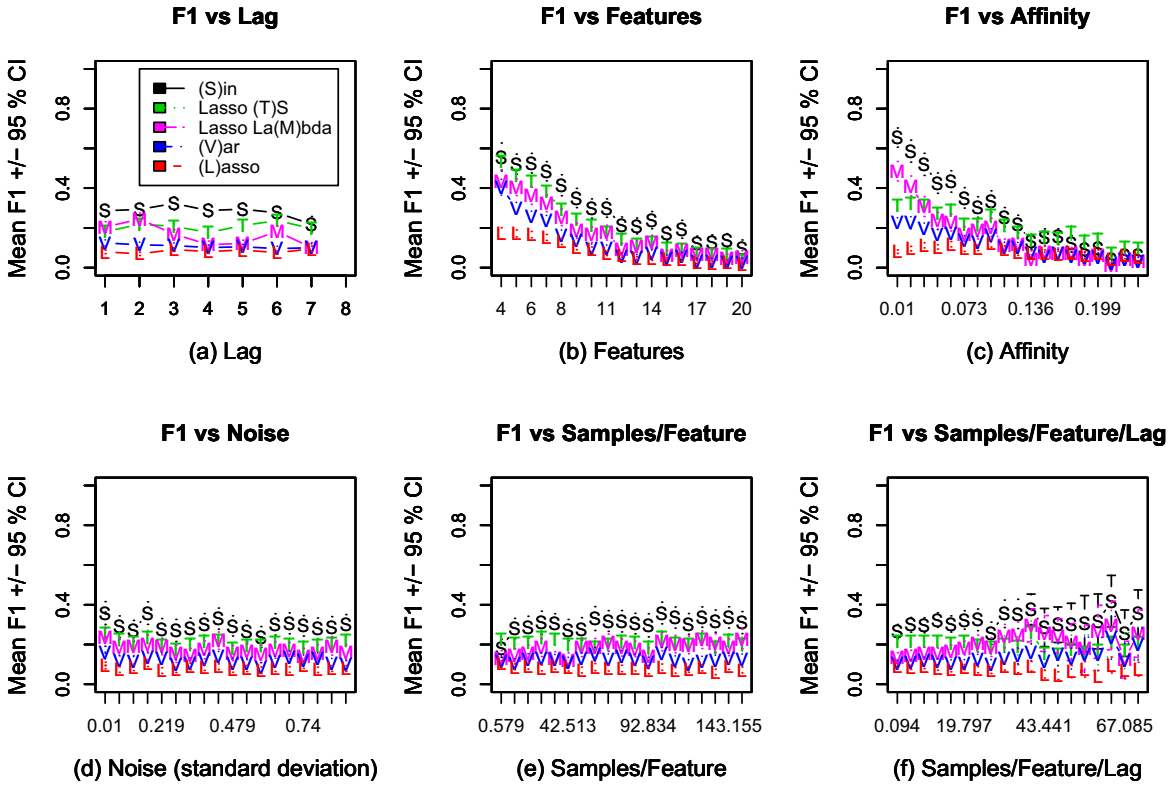

Figure 1: F_1 **varies as a function of (a)** Max_T**, (b)** *features*, **(c)** *sparsity*, **(d)** *noise*, **(e)** *samples per feature* **and (f)** *samples per feature per lag.*

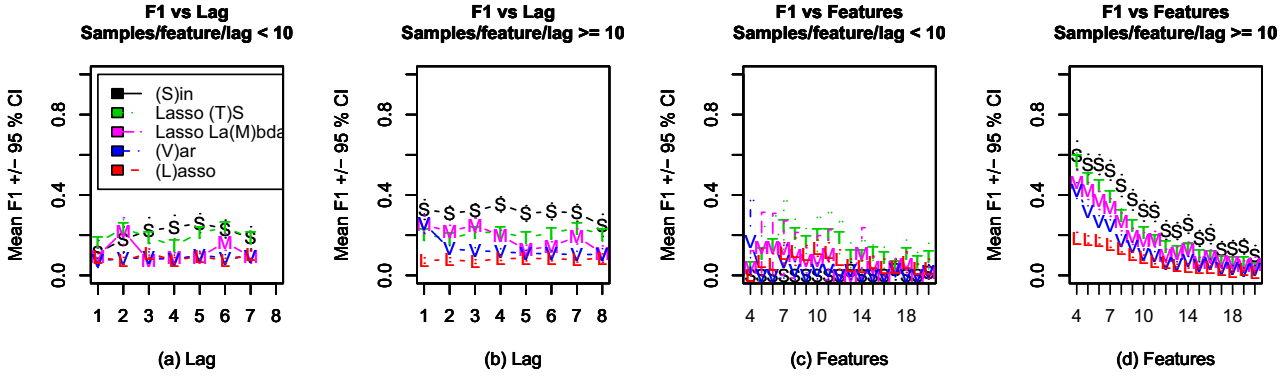

Figure 2: F_1 **varies as a function of** *maximum lag* (Max_T, **(a) and (b))** **and** *features* (p, **(c) and (d))** **conditioned on** *samples per feature per lag* ($n/p/Max_T$)**.**

to examine the overall trend of their relative performance. We then "drilled down" into some selected sub-spaces of the problem space to examine some specific questions. Part of this is done by examining the distributions of relative performance measures, conditioned upon some restrictions of one or more of the parameters considered, in terms of inequalities (e.g. the sample size per feature be greater than 10 or less than 10.)

The parameters of the problem space that we varied are: the number of features, the maximum lag, the number of samples per feature, the number of samples per feature per maximum lag, the maximum standard deviation of noise, the maximum/minimum coefficient of an effect, the affinity or anti-sparsity, which is defined as the probability of an edge presence.

In each of these series of experiments, the simulations were repeatedly run on a large number of different data sets, with all the reported performance numbers averaged over them,

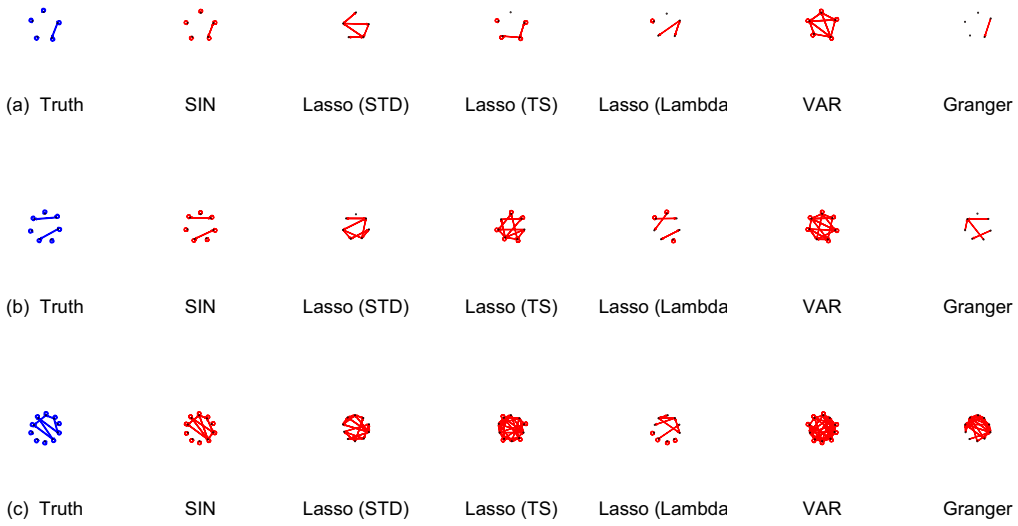

Figure 3: Comparison between the true, generative graph structure (left) and the graphs learned by various algorithms on multiple synthetic data sets (right).

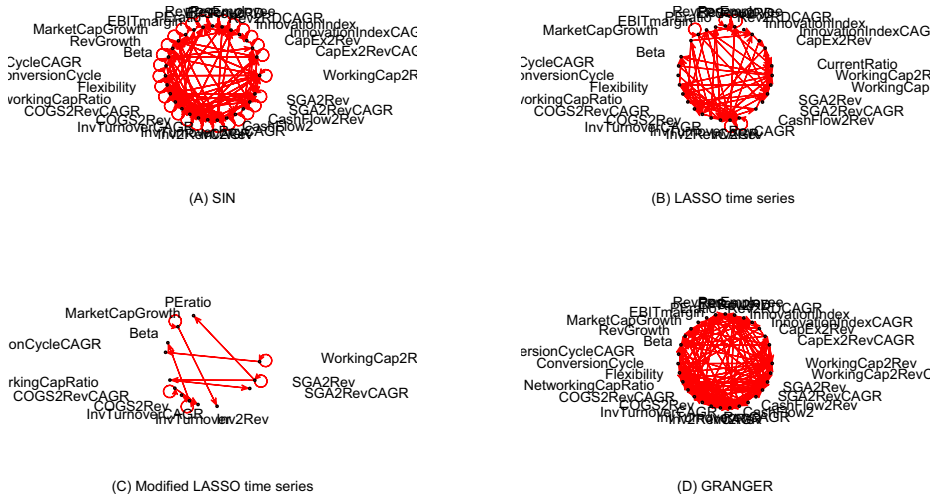

Figure 4: Graph structure output by various algorithms on real KPI data.

and confidence intervals calculated using the standard error. The results of these experiments are shown in Figure 1. Although complete results for exhaustive Granger are not shown (due of the excessive time required to complete each trial, as will be shown in the plots of running time: c.f. Figure 5), preliminary experiments indicate exhaustive Granger's performance is somewhere between SIN and VAR. Plot (a) indicates that increasing lag (Max_T), in and of itself, does not affect performance that much, while (b) shows the expected drop in performance as the number of features increases. This decrease in performance is also seen in (c) as a function of affinity. Interestingly, as the density increases, most of the algorithms suffer, while the time series lasso (T) maintains its performance and thus relatively improves. Plot (d) suggests that all methods are relatively robust to noise. Finally, plots (e) and (f) show a shrinking of the gap between SIN and lasso as the number of samples available to the algorithm (weighted by feature size and lag, respectively) approaches zero. This makes sense in light of the fact that the regularized lasso has many fewer parameters to estimate than the full VAR or SIN model, and thus can achieve comparable results with fewer examples.

Because of the marginalization over unmeasured parameters in the plots above, it is somewhat difficult to tease out exactly how variations in groups of parameters affect each algorithm's relative and absolute performance. Figure 2 attempts to address this issue by showing a detail of Figure 1, this time splitting the plots out by conditioning on sample size. F_1 vs maximum lag in (a) and (b), and F_1 vs p in (c) and (d). Plots (a) and (c) are for experiments with small samples per feature per lag ($n/p/Max_T < 10$), while (b) and (d) show trials with a relatively large num-

ber of samples per feature per lag. We can see that SIN dominates when it has access to lots of examples (b). As the number of examples decreases, however, the lasso-based methods become more competitive (a). A similar pattern is seen between plots (c) and (d). We see that SIN excels when given data containing few features and lots of data (d), and struggles otherwise. Again, due to the high computation time, Exhaustive Granger trials are not shown.

Figure 3 shows the actual graph structure output on a few such trials. The leftmost graph shows the true structure from which the synthetic data was generated. To the right are the graphs learned by each of the six algorithms. Examining these output graphs is illuminating, since they contain extra information that is lacking in simpler performance metrics summarized as a single number. For example, you can see, for this subspace of the problem space, SIN is very accurate, VAR tends to output dense graphs, Exhaustive Granger seems less robust alternately outputting dense and spare graphs, and the model selection mechanism of the time series Lasso with the "Lambda" modification appears to be working well.

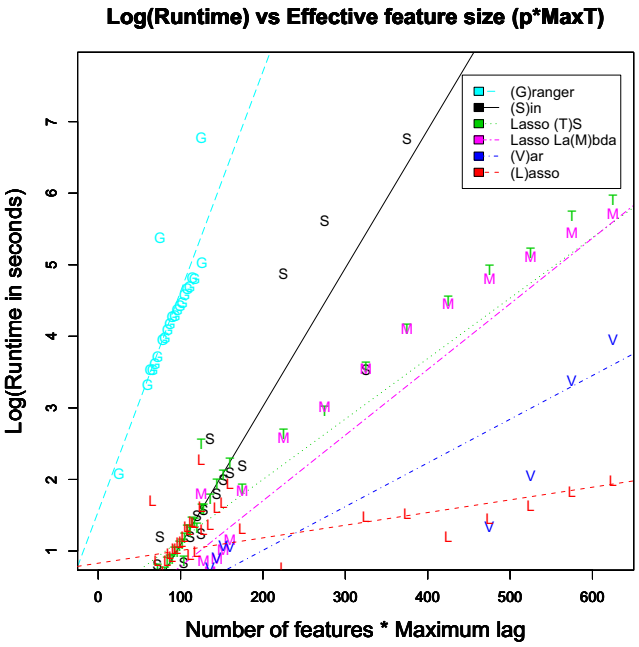

Figure 5: Log runtime (in seconds) as a function of effective feature size ($p * MAX_T$)).

Figure 5 shows the rate at which the *running time* of each algorithm grows as a function of the *number of features*, $15 < p < 125$, times the *maximum lag*, $1 < \text{Max}_T < 5$. Since all of these algorithms (except standard lasso) lag the data before attempting to model it, the effective number of features of the data is p*Max$_T$. Time is plotted on a log scale to better show the exponential growth in running time. Exhaustive Granger (denoted "Granger" in the graph) clearly is the most expensive algorithm taking over ten minutes for a dataset of less than 100 features. SIN grows at a slightly slower rate, followed by the two lagged lasso methods (T and M in the figure). Standard lasso grows the most slowly since it does not lag its input data. VAR is interesting because it starts out fast but begins to run much more slowly as Max$_T$ is increased. This is because, unlike the other algorithms, VAR actually searches the space of possible lagged models in order to choose the best fit.

5.2 Real World Data

In this subsection, we present some results of applying our causal modeling methods on a real world data set involving key performance indicators (KPI's) of electronics companies. The problem of monitoring and analyzing performance indicators of corporations is important in business investment decision making, and has recently received considerable attention [15, 16]. This particular data set was obtained from Standard and Poor's Compustat database [27].

The data set consists of values of various performance indicators for electronics companies that are in the industry group of "semiconductors and semiconductor equipments." Specifically, quarterly data over the duration of three years were pulled, for companies having at least 25 million dollars in annual revenue. The performance indicators in the data set include financial performance metrics such as Revenue growth, EBIT margin, productivity (Revenue/Employee), ROA, Market Cap Growth, Earnings per Share (EPS), PE Ratio, and Beta. The data also include lower level (operational) metrics such as Revenue per R&D Spend, Business Week's Investing 4 Future Index, Capital Expenditure / Revenue, Current Ratio, Working Capital/Revenue, COGS/ Revenue, SG&A Revenue, Operating Cash Flow/Revenue, Inventory Cost/Revenue, Inventory Turnover, Cash conversion cycle in days, SG&A Revenue, and Net Working Capital Ratio.

For many of these metrics, we consider both "absolute" values and the "CAGR" values or the "Compound Annual Growth Rate", which measures the annual rate of growth of the KPI in question. We note that some normalization and outlier filtering were performed in generating these data.

The results of running each of the structure learning algorithms on this KPI data are shown in Figure 4. Some interesting observations can be made, particularly with the output graph of "Modified Lasso time series". For example, we see that SGA2Rev (SGA to Revenue) is causally related to PE Ratio. We also see that Inventory Turnover is causally related to Beta. Since SGA to Revenue and Inventory Turnover are lower level operational metrics than PE Ratio and Beta, which are financial, this admits the interpretation that the former two metrics could be "levers" to pull for attaining desired financial performance in terms of the latter two. Although interpreting these relationships properly is difficult without deeper domain knowledge, it is clear that the graph learned by the lasso with lambda tuning is the most succinct and potentially useful for corporate performance management purposes.

Even with domain expertise, making sense of the very dense models generated by Exhaustive Granger (denoted "GRANGER"), SIN and Var would be tedious, if not impossible, and would likely yield few insights.

6. CONCLUDING REMARKS

We have presented a systematic evaluation of the relative performance of a host of related methods of temporal causal modeling based on Granger causality and graphical modeling. Our empirical evaluation has demonstrated that some of the new model selection methods in regression and

graphical modeling are effective for providing a practical alternative to canonical methods, which are more exhaustive in nature. In certain scenarios, it has been found, they can add extra predictive accuracy and may also help improve the interpretability of obtained models by arriving at more succinct models. In the future, efforts are required to further pinpoint the conditions in which the different approaches discussed in the paper are most effective, and the range of real world problems for which they add value. Various possibilities should also be explored to improve the performance of these methods, possibly by combining them with other techniques known in the causal modeling literature.

Acknowledgements

The authors would like to thank Saharon Rossett, Rick Lawrence, Markus Ettl and Rama Akkiraju of IBM Research, and Richard Scheines, Clark Glymour, and Peter Spirtes of CMU for discussions and support.

7. REFERENCES

[1] T. Chu and C. Glymour. Semi-parametric Causal Inference for Nonlinear Time Series Data. *J. of Machine Learning Res.*, submitted, 2006.

[2] T. Chu, D. Danks, and C. Glymour. Data Driven Methods for Nonlinear Granger Causality: Climate Teleconnection Mechanisms, 2004.

[3] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation*, 9:251280, 1990.

[4] A. Dobra, B. Jones, C. Hans, J. Nevins, M. West. Sparse graphical models for exploring gene expression data. *J. of Multivariate Analysis*, special issue on Multivariate Methods in Genomic Data Analysis, 90, 196-212, 2003.

[5] M. Drton and M.D. Perlman. A SINful Approach to Gaussian Graphical Model Selection. Department of Statistics, University of Washington, Technical Report 457, 2004.

[6] B. Efron, T. Hastie, I. Johnstone and R. Tibshirani. Least Angle Regression (with discussion). *Annals of Statistics*, 2003.

[7] M. Eichler. *Graphical modeling of multivariate time series with latent variables*. Preprint, Universiteit Maastricht, 2006.

[8] N. Friedman, I. Nachman, and D. Peer. Learning Bayesian network structure from massive datasets: The "sparse candidate" algorithm. In *(UAI'99)*, 1999.

[9] P.D. Gilbert. Combining VAR Estimation and State Space Model Reduction for Simple Good Predictions. *J. of Forecasting*: Special Issue on VAR Modelling, 14:229250, 1995.

[10] G. Golub, M. Heath, and G. Wahba. Generalized cross validation as a method for choosing a good ridge parameter. *Technometrics*, 21: 215-224.

[11] C. W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37: 424-438l, 1969.

[12] D. Heckerman. A Tutorial on Learning with Bayesian Networks, In *Learning in Graphical Models*, M. Jordan, ed., MIT Press, Cambridge, MA, 1999.

[13] P. O. Hoyer, S. Shimizu, and A. J. Kerminen. Estimation of linear, non-gaussian causal models in the presence of confounding latent variables. In *(PGM'06)*, pp. 155-162, 2006.

[14] M. Kalisch and P. Buehlmann. Estimating high-dimensional directed acyclic graphs with the PC algorithm. Technical report No. 130, ETH Zurich, 2005.

[15] R. Kaplan and D. Norton. The Balanced Scorecard - Measures that Drive Performance. *Harvard Business Review*. 71-79, 1992.

[16] R. Kaplan and D. Norton. *The Balanced Scorecard: Translating Strategy into Action*. Harvard Business School Press. Boston, MA, 1996.

[17] S. Lauritzen. *Graphical Models*. Oxford University Press. 1996.

[18] C. Meek. *Graphical Models: Selecting Causal and Statistical Models*. PhD thesis, Carnegie Mellon University, Philosophy Department, 1996.

[19] N. Meinshausen and P. Buhlmann. High dimensional graphs and variable selection with the Lasso. *Annals of Statistics*, 34(3), 1436-1462, 2006.

[20] A. Moneta and P. Spirtes. Graphical models for the identification of causal structures in multivariate time series models. In *Proc. Fifth Intl. Conf. on Computational Intelligence in Economics and Finance*, 2006.

[21] R. Opgen-Rhein and K. Strimmer. Learning causal networks from systems biology time course data: an effective model selection procedure for the vector autoregressive process. *BMC Bioinformatics 8* (suppl.) in press, 2007.

[22] J. Pearl. *Causality*. Cambridge University Press, Cambridge, UK, 2000.

[23] S. Roweis and Z. Ghahramani. A Unifying Review of Linear Gaussian Models, *Neural Computation*, Vol. 11, No. 2, 305-345, 1999.

[24] S. Shimizu, A. Hyvärinen, P.O. Hoyer, and Y. Kano. Finding a causal ordering via independent component analysis. *Computational Statistics & Data Analysis*, 50(11): 3278-3293, 2006.

[25] R. Silva, R. Scheines, C. Glymour, P. Spirtes. Learning the structure of linear latent variable models. *J. of Machine Learning Res.*, 7(Feb):191-246, 2006.

[26] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. MIT Press, New York, NY, second edition, 2000.

[27] Standard & Poor's Compustat Data. Available from *http://www.compustat.com*.

[28] R. Scheines, P. Spirtes, C. Glymour,and C. Meek". *TETRAD II: Tools for Discovery*. Lawrence Erlbaum Associates, Hillsdale, NJ. 1994.

[29] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal. Statist. Soc B.*, Vol. 58, No. 1, pages 267-288, 1996.

[30] P.A. Valdes-Sosa *et. al.*. Estimating brain functional connectivity with sparse multivariate autoregression. *Philos Trans R Soc Lond B Biol Sci.* 2005 May 29;360(1457):969-81, 2005.